



## **On Failure in Managed Enterprise Networks**

Daniel Turner, Kirill Levchenko, Jeffrey C. Mogul, Stefan Savage and Alex C. Snoeren

HP Laboratories

HPL-2012-101

### **Keyword(s):**

networking; failure; reliability; measurement

### **Abstract:**

Enterprises must maintain and improve the reliability of their networks. To do this at reasonable expense, many enterprises choose to outsource the management of their network to third parties, who care for large number of networks for a variety of customers. We obtain and analyze almost a year's worth of failure data for thousands of enterprise networks from such an enterprise service provider. Our data includes device logs, network probes, call center tickets, and incident reports. We study not just the causes and frequency of failures, but also their perceived impact and how failures are resolved. We also characterize what types of failure cause end users to complain. Finally, we show that low-level data may be insufficient to accurately characterize user-visible failures.

External Posting Date: May 9, 2012 [Fulltext]  
Internal Posting Date: May 9, 2012 [Fulltext]

Approved for External Publication

# On Failure in Managed Enterprise Networks

Daniel Turner, Kirill Levchenko, Jeffrey C. Mogul<sup>†</sup>  
Stefan Savage and Alex C. Snoeren

UC San Diego and <sup>†</sup>HP Labs

## Abstract

Enterprises must maintain and improve the reliability of their networks. To do this at reasonable expense, many enterprises choose to outsource the management of their network to third parties, who care for large number of networks for a variety of customers. We obtain and analyze almost a year’s worth of failure data for thousands of enterprise networks from such an enterprise service provider. Our data includes device logs, network probes, call center tickets, and incident reports. We study not just the causes and frequency of failures, but also their perceived impact and how failures are resolved. We also characterize what types of failure cause end users to complain. Finally, we show that low-level data may be insufficient to accurately characterize user-visible failures.

## 1. INTRODUCTION

Communications networks are absolutely essential to modern business processes, yet are also tremendously complex and potentially fragile. Thus, the operational state of a network can never be taken for granted, but must be carefully monitored and managed reactively, to handle problems as they arise. However, efficient and effective network management is hard, requires specialized skill sets, and can be costly to provide, especially at small scales. For these reasons, among others, many enterprises outsource the management of key components in their network to third-party providers. Such “enterprise service providers” (ESPs) can exploit their ability to hire domain experts, build up domain expertise, amortize investments in a range of monitoring and assessment tools, and, perhaps most important, to afford a 24-hour staff who can respond to incidents whenever they occur.

However, there is little public information available about the underlying problems faced by such managed enterprise networks, or the challenges this arrangement imposes on network managers themselves. While the research community has recently started to characterize empirical failure data for individual ISP, datacenter and campus networks [8, 14, 5, 2], those networks are quite different in character—enjoying greater levels of homogeneity in design, construction, and configuration. By contrast, managed networks

come in all shapes and sizes, with a wide variety of components, topologies, demands, and histories. Indeed, while some customers architect their managed networks cooperation with their ESPs, in many (if not most) cases, ESPs inherit these networks long after they have been constructed, and ESP managers must accommodate the full complement of third-party vendors, Internet service providers, and configurations that come with these legacy designs.

This paper provides a first step in empirically characterizing the managed enterprise network environment—as seen from the viewpoint of a particular enterprise service provider. Using a combination of low-level network event data (e.g., syslog, SNMP, NetCool, etc.) and a full trouble-ticket database, maintained by network managers, we analyze the full range of problem episodes that took place over almost an entire year. This data covers thousands of individual customer networks. We especially looked at the relationship between low-level network events and “unresolved problem” (URP) minutes – periods when a customer network is experiencing some problem (which can vary from minor requests for configuration changes to complete loss of connectivity).

Based on our analyses, our paper makes these principal contributions:

- *Limitations of low-level data sources.* Traditionally, many researchers—ourselves included—have evaluated network problems as seen primarily from low-level data sources such as syslog, SNMP, or active probes (pings). However, we have found that attempting to unify such data with the “ground truth,” as represented by trouble tickets, is extremely challenging due to differences in abstraction, lack of naming normalization, and limited vantage points. We conclude that today’s low-level data sources alone are unlikely to provide a comprehensive characterization of failure as experienced by users or network managers.
- *Characterization of critical problems.* A frequent problem in analyzing network problems is distinguishing between disruptions that are important and those that have little impact (or perhaps were even planned). However, since we analyze failures from the standpoint

of the ESP (via their ticketing system), we can explicitly distinguish between high-severity issues and those that have less-immediate impact. We found that the principal source of high-severity unresolved problem minutes are connectivity failures that result from third-party ISP issues (i.e., problems that only ISPs can resolve); thus, significant further improvements in severe problems can only be solved via additional diversity in WAN links.

- *Characterization of non-critical problems.* We similarly show that the majority of problems (as represented in tickets) and in total problem minutes are due to lower severity issues (e.g., problems with particular LAN switches, requests for configuration changes, etc.), and that these are the dominant source of work for the ESP we studied. We characterize these problems, and show that, unlike high-severity problems, they cover a far more diverse set of issues, yet the ESP can usually fix these problems without assistance from other vendors or providers.
- *Customer involvement.* Our database includes both incident tickets as well as call records (i.e., when a customer reports a problem). Because these two kinds of records are cross-linked, we can distinguish between problems that drive customer calls, and problems to which customers are oblivious. Unsurprisingly, we find that various kinds of connectivity problems typically generate customer calls (although we find that certain classes of outages, which we believe represent failures of redundant links, are far less likely to be reported). Interestingly, however, call volumes are not heavily concentrated in any particular category, suggesting that ESPs have sufficient monitoring capability to discover most classes of problems without customer-generated reports.
- *Customer variation.* Finally, we looked at the variation among customers. We found, with the exception of high-severity problems, that the breakdowns by problem type, source of problem, and resolution can vary tremendously between customers; this illustrates that studies focusing on only a single network are unlikely to capture the full breadth of problems that an ESP must handle.

## 2. RELATED WORK

Complex artifacts inevitably experience failure, and network designers have always had to anticipate and manage failure. Indeed, failure tolerance is a key motivation for much of networking research, from routing protocols to content distribution networks. However, to manage failure effectively, one must understand the prevalence of different failure modes. The research community has made a range of contributions in service of this goal.

First, there is a broad body of work focused on inferring network outages, based both on indirect, external measurements (e.g., [4, 3, 6]), or on limited, internal measurements (e.g., [1, 9, 15, 12]). Second, a small number of detailed characterization studies have used more comprehensive data, but focusing on a particular network. Among these, Shaikh *et al.* have examined failures through the lens of OSPF in a large enterprise network [11], Markopoulou *et al.* characterized failures in the Sprint IP backbone, also using routing data [8], Turner *et al.* provide a similar analysis of the CENIC backbone using syslog and trouble ticket data, and most recently both Benson *et al.* [13] and Gill *et al.* [5] have provided empirical studies of data center networks. (Gill *et al.* in particular focus on failures using syslog, SNMP, and traffic flow data.) Perhaps unsurprisingly, there are considerable differences in failure modes across these different kinds of networks; our own findings echo this variation.

Our work is distinct from previous efforts not in any particular measurement approach, but in both its focus on *managed enterprise networks*, and in the large number of customer networks covered by our study. Finally, we exploit the large feature set in the trouble ticket database to analyze issues, such as problem severity and customer involvement, that are difficult to characterize in any other way.

## 3. THE NETWORKS

Unlike a backbone or datacenter environment, where typically a single operator deploys and maintains the network, enterprise networks are often made up of a variety of independent network sites, spread across different physical locations, that may each have been deployed and maintained by distinct administrators, but logically connected with VPNs or other wide area links. Moreover, due to their vast differences in scale and scope, as well as the resources and business needs of the companies deploying them, enterprise networks are less likely to track current best-practices solutions, but instead exhibit significant heterogeneity. Hence, one of the most challenging aspects of studying enterprise networks is simply understanding their topological makeup.

Due to this inherent diversity, any study that focuses on one particular company or location is unlikely to accurately characterize the full spectrum of issues that arise in enterprise networks. By partnering with an enterprise services company, we are able to gain insight into myriad different networks. The downside, however, is that in the majority of cases, the manager does not own or operate the networks. Instead, it is contracted only to monitor, manage, and—in some cases—repair these networks, and often only has control and visibility over a subset of any given network.

### 3.1 How we obtained the data

We were fortunate to have the cooperation of HP Enterprise Services (HP ES). One of the many services that HP ES offers to customers is to manage certain aspects of a customer's network infrastructure. The role of HP ES

varies considerably among these customers, but typically includes taking responsibility for responding to alarms raised by customer-owned network devices (routers and switches), responding to issues reported by customers (via phone or email), handling problems with WAN suppliers, and resolving outages related to devices and links under management by HP.

Note that the discussion that follows is based on information that we received from HP ES several years ago.

It is rare for HP ES to have end-to-end responsibility for any customer network, and so we have chosen to present our results in terms of customer “sites” rather than customer networks. This also means that, because HP is not typically responsible for end-to-end customer network connectivity, HP ES does not capture data that directly measures such connectivity.

In some cases, the customer has full responsibility for designing the network, and thus HP has no responsibility for preventing failures, only for responding to them. In other cases, HP does work with customers to manage devices so as to avoid future failures, or to advise customers on techniques such as adding redundancy, upgrading systems, re-routing traffic, etc. However, we do not have access to the relevant system level agreements (SLAs), so we cannot distinguish between customers on this basis, and one cannot infer from our data sources whether either HP or a customer was in any way “at fault.”

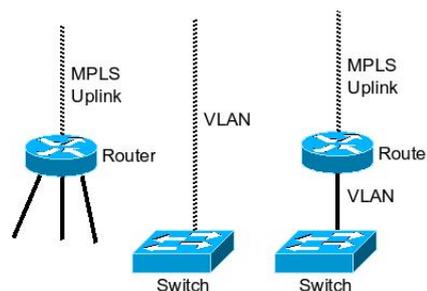
HP takes customer data privacy very seriously. Because the individual data records refer to specific customers and their network configurations, we cannot provide any of the raw data to others (including any co-authors of this paper who are not current HP employees). The results we present, of necessity, can only reflect aggregate statistics. Also, as a condition of getting access to this data, we must avoid revealing specifics of HP’s business, such as the number of customers or the growth rate in customers. All that we can say is that we have data from thousands of customer sites.

## 3.2 Topologies

HP sells service for hire and, as such, customers choose which of the devices in their network it makes financial sense to have managed. Some customers have their entire site managed, while others elect only to monitor an individual device, such as an edge router. Due to privacy concerns, we did not contact customers directly about their networks. Hence, we are able to obtain information regarding only the managed portion of the enterprise networks.

### 3.2.1 Information sources

HP maintains a managed device list (MDL) that contains every network device it manages, as well as various meta-data about each device. The meta-data includes customer ID, deployment location (site), model number, and functionality (router, switch, base-station, etc.). Because there is churn in the devices under management, we received a small number



**Figure 1: The three most common network topologies. Black lines that are connected to only one device represent links to devices not managed by HP.**

of updated MDLs throughout our study.

Managed devices may connect to a variety of other network devices, only a subset of which are themselves managed. Hence, it is important to understand the underlying network topology. To help us understand network topology we were provided with a managed interface file (MIF) describing important interface in a customer’s network. The MIF is built by probing managed devices via SNMP and contains an entry for interfaces that meets a specific criteria. For those interfaces with an IP address, the MIF lists the IP address and subnet. It also contains a text string that provides a reference to the layer-two connectivity of the interface. As with the MDL, during our study we received a small number of updates to the MIF.

Because the MIF is the result of an automated analysis tool, its results are not perfect. For example, interfaces connected to IP telephony devices, which are relatively unimportance, are highly likely to match the automated filter’s importance criteria. Conversely, if HP does not have SNMP access to a device no interfaces will be auto-discovered, though there is a facility for interfaces to be manually added. Finally, due to a conservative set of rules for removing an interface from the MIF, some devices will appear to have managed interfaces for a few days after being removed from the MDL.

### 3.2.2 Reconstruction methodology

Armed with the managed device and interface lists, we reconstructed the topologies of the managed (portion of) customer networks at each site. This exercise is necessarily a multi-step, best-effort, data-driven process. We begin by looking for point-to-point links, this is done by pair-wise comparison of interfaces that have an IP address and netmasks as specified by the MIF; because most interfaces do not have an IP address this approach identifies only a small number of links. Next, we looked at the free-text “connectivity” field of an MIF entry, which sometimes indicates what each interface is plugged into. Often this is a string like “x.network.com [1[2]].” The first part of this string is easy to interpret: it informs us that this interface connects to a port on a device named x.network.com. The second

part identifies the line card and port, but the automated tool is often unable to determine this information and instead used default values. As a result, we only attempted to understand connectivity between devices, and ignored the specific ports in use. When neither an IP address nor connectivity string is present, we cannot determine to what the port is connected.

Beyond discovering links in each site, we also tried to understand the purpose of each interface. Towards this end, we classified each interface into one of four categories, by using textual clues in the interface “connectivity” string, —uplink, link to other network devices, VLAN, and other—based upon both the textual description and the IP configuration information. (For example, we consider an interface with a /252 netmask to be a point-to-point link.) The “other” category corresponds to interfaces whose purpose we cannot determine, or that appear to be plugged directly into an end host.

### 3.2.3 *Managed topologies*

To provide context for some of our findings regarding failure, we briefly characterize the managed network topologies. Perhaps surprisingly, the median network has only two managed devices (the average is 8). While it is possible that the median enterprise network is actually that small, we deem this unlikely. Rather, we suspect that HP frequently manages only a subset of its customers’ networks—likely, the edge and most-critical devices. To illustrate our contention, Figure 1 presents the three most common network topologies in the data-set. For example, in the left-most topology of Figure 1 there are two directly-connected hardware devices: a router and a switch. The router has an MPLS uplink, and is also connected to a VLAN. The switch is connected to the same VLAN—most likely over the physical link they share. Hence, we suspect that the router is an edge router, and that the switch is connected to servers and/or other network hardware not being managed by HP.

To give some idea what role these devices play in customer networks, we also used a set of heuristics to attempt to classify the managed interfaces of each device. To avoid drawing unsupported conclusions, our classification is deliberately conservative, leading to a modest classification rate of only 53%; i.e., we choose not to label 47% of the interfaces where there is some ambiguity regarding their role. Of the interfaces we do classify, 45% are VLANs, though we suspect this number to be overly represented due to one set of criteria for determining if an interface should be placed in the MIF. The next largest category, accounting for 32% of all interfaces, consists of interfaces connected to a server or network device HP does not manage. 12% of the interfaces are point-to-point links of unknown use (i.e., they may connect redundant switches, or serve to bridge remote customer sites, or something else entirely), while 7% of the interfaces are clearly uplinks (many of which are themselves point-to-point, but are not double counted).

While the median network size is only 2, many networks

are much larger, suggesting HP is managing the entire site. To give a flavor of these networks, we arbitrarily consider only those networks with at least ten managed devices. The median size of this restricted set jumps to 19 devices, with a mean of 34. On average, almost half of these devices are switches, followed by a quarter being wireless base-stations, 8% are routers, 5% firewalls, and 2.5% are power (e.g., uninterruptible power supplies with Ethernet interfaces). Surprisingly, however, the median percentage of wireless devices per site, at 48%, is actually slightly larger than the median percentage of switches per site, 44%.

Due to our inability to make definitive statements regarding what fractions of its customers’ networks HP ES manages, or (in many instances) about the role played by the interfaces, we choose not to present device failure data in terms of topology. In what follows, we treat each failure equally, regardless of whether it occurs at an edge router, wireless base-station, internal switch, etc. We note, however, that in many instances the topological importance of a device will impact the severity of its failure. E.g., a customer’s SLA is likely to specify that the loss of its edge router is a high-priority event.

## 4. DATA SOURCES

Traditionally, network failure studies have relied upon low-level logging information generated by automated monitoring facilities or by the network devices themselves — e.g., syslog, SMNP, and active probing. While these are frequently rich sources of information, they are fundamentally limited to recording symptoms, e.g., lost connectivity or high delay variance; they cannot serve as an authoritative source for failure causes, nor, generally, do they reveal how the root problem was addressed. Finally—and most importantly—they cannot speak to the impact of a failure: did it matter to the end user, and, if so, to what degree?

In contrast, this study considers data not only from traditional network monitoring sources, but also from a professional enterprise services firm’s trouble ticketing system. Hence, we are able to discuss not only the symptoms of failure, but also their causes, impacts, and resolutions. Surprisingly, we also find that the low-level data alone is decidedly insufficient to characterize the vast majority of incidents reported in the trouble ticketing system.

HP ES provided us with several data sources that they use to identify and resolve network incidents in their customers’ networks. Our data set begins in June 2010 and continues into April 2011. In this section, we explain the processes that generate the data, its business purpose, and relevant details of its makeup. We conclude by documenting our difficulty reconciling low-level data sources with the trouble tickets.

**Table 1: Top five syslog message types in archives.**

Syslog Message Type	% of all messages
LINEPROTO-5-UPDOWN	9.9%
CDP-4-DUPLEX_MISMATCH	8.6%
LINK-3-UPDOWN	8.2%
DOT11-6-ASSOC	4.7%
ILPOWER-5-IEEE_DISCONNECT	4.5%

## 4.1 Automated data sources

HP ES employs a number of automated systems to monitor the health of customer networks. We obtained data generated by both passive and active mechanisms.

### 4.1.1 Passive data collection

Typical of well-managed networks, most of the devices managed by HP are configured to send syslog and SNMP traps over the Internet to a central repository. These messages are passed through sophisticated alarm systems that correlate reports from several data sources to detect problems. Technicians use these alarm systems to help spot and troubleshoot network incidents.

**Syslog.** We were granted access to the repository of syslog messages. Unfortunately, most hardware devices are configured to send syslog on every interface, managed or not. As a result, a significant fraction of syslog events describe interfaces outside of the managed topologies. Hence, we filter all messages from devices and/or pertaining to interfaces that are not managed.

Table 1 lists the five most common syslog message types (after filtering out unmanaged devices and interfaces). These five event types represent 36% of all messages in syslog, and no other event type accounts for more than 4.5% of the entries. The second and fifth most popular message types—duplex mismatch warnings and power over Ethernet (PoE) disconnections—are not particularly interesting events. However, the first and third most frequent message types are layer-two link-state messages, which are far more likely to correspond to failure events. For the purposes of this study, we choose to limit our exploration to these up/down messages, due to their clear meaning, relatively high frequency, and natural pairing, which makes it easy to delineate the beginning and end of each incident.

Unfortunately, our topology reconstruction methodology leaves us unable to resolve both end points of many links. Rather than mix interfaces and links in our study, we consider interface state instead of link state. As in previous studies [14], we determine interface state by treating the syslog archive as an ordered but lossy log, and then correcting for lost messages.

**SNMP.** We were unable to obtain access to an unfiltered copy of SNMP traps. Instead, our view of SNMP traps is limited to those that successfully pass through a series of automated relevancy filters, implemented in the IBM/Tivoli

Netcool operations management software. The nature of these filters is obviously proprietary, so we are not at liberty to discuss its functionality. However, for the purposes of this study it suffices to assume that the filtering is effective at separating the wheat from the chaff—i.e., it represents the operators’ considerable expertise about which SNMP traps are meaningful, vs. those that are typically less informative.

### 4.1.2 Active measurements

While HP may not be responsible for end-to-end connectivity at each site, they still find proactive connectivity tests to be valuable in early incident detection. To this end, all devices managed by HP are pinged every 5 minutes, from HP’s network operations center. (For various practical reasons, HP cannot send pings from customer sites, and hence this data set does not represent customer-relevant connectivity; it is used only to detect device failure.)

As with SNMP data, our access to the active measurement records are filtered through Netcool. In their experience, HP ES determined that packet loss over the Internet, as well as highly utilized LAN links, leads to an unacceptable number of false positives if individual probe failures are reported. Therefore, Netcool only generates a record if two or more successive probes sent to the same interface fail. Even when considering instances with two successive probe failures, however, it is important to realize they might not indicate an actual incident, nor an incident for which HP has responsibility to detect and/or resolve.

## 4.2 Service tickets

Our final—and most informative—data source is the set of trouble tickets generated by the network operators themselves. We were provided with two different types of tickets: call center tickets and incident tickets. Call tickets are opened when a customer contacts a service desk to report an incident or submit a request. Incident tickets, created by network technicians, are opened either in response to an alarm or a call ticket. Incident tickets are intended to contain the full record of each technician working on a problem, including the resolution. While there are hundreds of potential fields in a call ticket, including many that are free-form text, we find ten to be especially useful and relevant. These fields are listed in Table 2, along with a brief explanation. Beyond these ten fields, we also extract any incident IDs from call tickets and vice versa, so that we can cross-reference them. Operators have systems in place to minimize duplicate tickets and strive for consistency across tickets, but as with any manual process there will be some mistakes.

## 4.3 Merging low-level events and tickets

Our first instinct when presented with these two distinct classes of data was to try to unify them, so that we could leverage the totality of information regarding failure events. Unfortunately, as we describe below, doing so proved surprisingly problematic—so much so that we were forced to

**Table 2: The 10 fields in HP ES service tickets that we consider here.**

Field name	Description
<b>Record ID</b>	A unique ID for each ticket that also encodes call or incident ticket
<b>Severity</b>	Level of degradation, from level 1 (complete) to level 5 (none), as determined by customer.
<b>Sub-category</b>	LAN, WAN, Hardware, Software, Application, ...
<b>Problem type</b>	Other, Error Message, Failure / Down, Connectivity, Break/fix ...
<b>Product type</b>	Other, Midrange, Server, LAN, ...
<b>Resolution code</b>	Fix/repair/correct, No problem found, Restore, Restarted, Power restored
<b>Customer</b>	Whom does this incident impact
<b>Downtime start</b>	First time at which customer was impacted
<b>Downtime end</b>	Last time at which customer was impacted
<b>Ticket open time</b>	Time at when ticket was opened (may be later than downtime start)

**Table 3: The frequency with which incidents of various causes are visible in our low-level data sources. All values are percentages of the entire set of incident tickets, not just that cause or matching set.**

Problem Type	% of tickets	Contain device name (%)	Match syslog (%)	Match Netcool (%)	Match both (%)
Overall	100	22.6	1.1	14.1	0.89
Other	17	4.2	0.26	2.6	0.2
Break/fix	14.6	2.7	0.07	1.7	0.05
Slow/no response	14	5.6	0.1	4.4	0.1
Failure/down	12	1.8	0.3	1.3	0.25
Connectivity	7.7	1.4	< 0.01	0.3	< 0.01
Error Message	5.9	1.5	0.08	1.2	0.07
Configuration	4.5	0.5	< 0.01	0.1	< 0.01

abandon the attempt, and instead report on each data source individually. Fundamentally, the challenge lies in the fact that there are many different ways to describe network devices and interfaces—IP address, MAC address, DNS name, customer-specific asset tracking tag number, etc.—and no universally accepted standard or practice regarding which one or ones are used.

#### 4.3.1 Methodology

As a starting point, we tried to leverage temporal correlation, as is commonly done in the literature [7, 14]: We matched the downtime start field in incident tickets to the start of an event in syslog. Due to the large number of syslog entries, attempting a purely temporal match yields a very large false match rate. We then further refined our matching procedure to require not just a temporal match, but also that the customer field of the incident ticket match the owner of the device sending the syslog message. While this improved the false match rate, it did not reduce it to a level where we feel confident in using the matches. Moreover, such a customer-based classification is frustrated by the fact that many customers call in to report trouble that turns out to be caused by devices owned by an ISP or hosting center. Hence, a company match is neither necessary nor sufficient.

In our third and most restrictive refinement, we again began with temporal matching, but considered only matches

between incident tickets that explicitly name a network device and syslog entries generated by or about that device. This method reduces the false match rate to a manageable level, but is clearly overly restrictive. In particular, it ignores the 77% of tickets that do not explicitly name a network device listed in the MDL.

#### 4.3.2 Results for various matching methods

Table 3 summarizes the trouble incidents that we are able to match to a low-level data source. We break down the incidents in terms of problem type (which is one of the ticket fields listed in Table 2); we show the seven most common problem types. For each distinct problem type, we show the frequency of this type (regardless of whether we found a match or not), the fraction that exactly mention a device name in the MDL (an upper limit on the fraction we can expect to match), the fraction that matched a syslog up/down event lasting one minute or longer, the fraction that matched either an SNMP trap or active probe record that passed the Netcool filter, and the fraction that matched both.

It is immediately apparent that only a very small fraction of the incident tickets can be matched to a low-level data source. Even considering just the <23% of tickets with device names, only 62% of them match Netcool alarms, and less than 5% match syslog events. Most incidents that match in syslog also appear in Netcool, however, so the low-level

**Table 4: Product type for the 14% of incidents that matched a Netcool record.**

Product type	% of tickets	Median duration (HH:MM)	% of URP minutes
router	3.751	10:10	4.110
switch	2.658	18:38	2.738
router/switch	1.399	04:00	0.179
firewall	1.264	07:57	0.764
Not specified	0.531	07:42	0.234
Loadbalancer	0.529	1 day	1.101

data sources appear self-consistent. Given that there are  $3\times$  more syslog failure events that last longer than one minute than there are incident tickets, however, we find the overall matching rate surprisingly poor. Moreover, as the table clearly shows, low-level logging tools do not appear to be significantly more effective at detecting any types of problems than others.

One might instead hypothesize that certain devices are more likely to generate syslog or SNMP traps, and hence more readily match incident tickets. Unfortunately, as Table 4 shows, this is not the case. Of the 14.1% of incidents with a matching Netcool record, the product type is fairly well distributed across the top 10 device types. Moreover, the matching incident durations are not significantly different from the overall incidents. Similar studies across other fields, including severity, resolution code, and duration, indicate that the low-level data sources are uniformly poor indicators of which incidents appear in the trouble ticketing system.

We conjecture that the reasons for this are varied. First, even in a perfect world we would not expect 100% matching: many of the incidents in the ticketing system are unrelated to layer-2 link failures, so would not appear in the syslog data. For example, 16% of the incidents that match a Netcool record are in the “network software” sub-category. Conversely, Netcool alarms unavoidably contain false positives: over 10% of the incident tickets matching Netcool alarms contain the resolution code “no problem found” or “no fault found.”

More generally, we observe that requiring an exact match on network device name is extremely problematic, given the various ways in which devices are referred to in trouble tickets and named in low-level data. We conjecture that the matching would have been much more effective if there was a canonical naming scheme for devices across the various data sources. Yet removing this restriction—in addition to creating a huge number of false positives—leaves a large number of incidents unmatched, which indicates a significant portion of the incidents simply are not detected by these low-level data sources. This is likely due to the restricted nature of the management relationship, where customers are impacted by the complex relationship between many devices, not all of which HP manages or monitors.

**Table 5: Classes of incidents (by severity) in customer networks managed by HP.**

Class	Severity			
	1	2	3+	All
WAN	18.5%	2.9%	4.7%	4.8%
LAN	17.5%	3%	11.5%	11.9%
Hardware	31%	62%	37.3%	39%
Software	4.5%	3.7%	16.9%	15%
Config	4.3%	3.1%	6.1%	6%
Connectivity	19%	14.9%	11.2%	11.4%

Given the extremely limited coverage of the low-level data sources, we choose to ignore them going forward, and instead present results based entirely on the trouble tickets. We are far more confident these tickets represent incidents that customers really do care about. Moreover, we conclude that—at least in the managed enterprise environment—one should view with a healthy degree of skepticism any studies of network failure that are based entirely on low-level data sources, without any ability to validate them against some form of ground truth.

## 5. RESULTS

To protect the privacy of HP’s customers we can not report raw numbers. Instead we concentrate on three normalized metrics. First, we report the fraction of all incident tickets in a particular condition (sometimes qualified by other factors such as incident severity). The second metric we report is “unresolved problem” (URP) duration – the period when a customer network is experiencing a problem, reported as the median for a ticket category. Note that unresolved problem duration does not necessarily translate to downtime, as some tickets simply reflect reduced redundancy and others do not impact connectivity at all. Finally, we report the percentage of all unresolved problem minutes that can be attributed to a particular incident category (this is to capture the presence of long tails, not well described by the median).

### 5.1 Incident characterization

The ticket system contains incidents involving HP leveraged equipment—equipment owned and operated by HP to provide connectivity for its customers—as well as customer-owned equipment managed by HP. Because these two classes of incidents have fairly different characteristics, we treat them separately.

#### 5.1.1 Customer equipment incidents

We begin by characterizing network incident tickets involving customer networks managed by HP, which account for 95% of all network incident tickets. Table 5 shows the breakdown of these tickets by problem class and severity. Problems in the WAN and LAN dominate the highest-

**Table 6: Resolution of incidents (by class) in customer networks managed by HP.**

Resolution	WAN	LAN	HW	SW	Config	Connectivity	All
Fixed	50%	42%	35.2%	51%	50.2%	42%	41%
No prob. found	5.2%	10.5%	7.8%	15.5%	4.7%	5.1%	10%
Power restored	1.2%	2.5%	11.6%	2.1%	0%	6.5%	6.4%
Replaced	1.5%	0%	1.1%	1.1%	3.6%	1.3%	1.2%
3 <sup>rd</sup> Party	6.3%	3.2%	4.6%	0%	1.4%	1.4%	3%
Reboot	1%	1%	1.2%	0%	0%	3%	1%

**Table 7: Resolution of incidents (by severity) in customer networks managed by HP.**

Resolution	Severity			
	1	2	3+	All
Fixed	48.8%	35.9%	41%	41%
No prob. found	1.5%	5.9%	10.5%	10.1%
Power restored	11.8%	18.8%	5.6%	6.4%
Replaced	1.5%	1.9%	1.6%	1.2%
3 <sup>rd</sup> Party	8%	6.8%	2.6%	2.9%
Reboot	1.5%	1.9%	1.1%	1.2%

severity incidents, accounting for 36% of severity-1 incidents, with network hardware the next largest contributor at 31%. Network hardware accounts for an even larger proportion of incidents in the lower-severity (3–5) categories, with the proportion of software-related problems nearly four times larger than in the highest severity category. On the other hand, WAN is more heavily represented in the highest-severity category; this should not be surprising given its potentially broader effect.

**Resolution.** How are network incidents resolved? Table 6 shows the four most common ways in which problems are resolved. The majority of most problem classes are simply fixed, but LAN and hardware seem more problematic. 11.6% of hardware problems were solved by simply restoring power, while LAN has an unusually large percentage of cases where an unusual resolution (i.e. not one of the top four classes) was employed. Software has disproportionately frequent occurrences of “no problem found,” likely due to the devilish difficulty of reproducing software bugs.

Of course, customers are likely not particularly concerned with the class of problem; presumably they care far more about its severity. Table 7 shows the relationship between incident resolution and severity. Happily, almost half (48.8%) of severity one incidents are fixed. A significant fraction, however, required a third party to address. In particular, 11.8% were resolved by restoring power, and another 8% were addressed by the vendor. Severity two incidents have a very similar profile, but things start to change with the lower severity incidents. In particular, HP was unable to reproduce the problem in 10.5% of the cases, and the tail of resolution

**Table 8: Median duration of incidents (by class and severity) in customer networks managed by HP.**

Class	Severity			
	1	2	3+	All
WAN	01:54	04:05	08:36	07:38
LAN	02:45	03:29	24:00	24:00
Hardware	02:13	03:38	13:08	11:13
Software	03:08	02:51	24:00	24:00
Config	02:00	01:41	24:00	23:09
Connectivity	02:45	03:28	04:26	04:18

**Table 10: Median duration of incidents (by resolution and severity) in customer networks managed by HP.**

Resolution	Severity			
	1	2	3+	All
Fixed	02:32	05:04	14:52	13:01
No prob. found	00:25	01:20	23:07	21:28
Power restored	02:22	02:00	03:36	03:19
Replaced	02:22	24:00	6 days	4 days
3 <sup>rd</sup> Party	04:36	06:24	13:40	11:46
Reboot	1:53	01:13	04:12	03:08

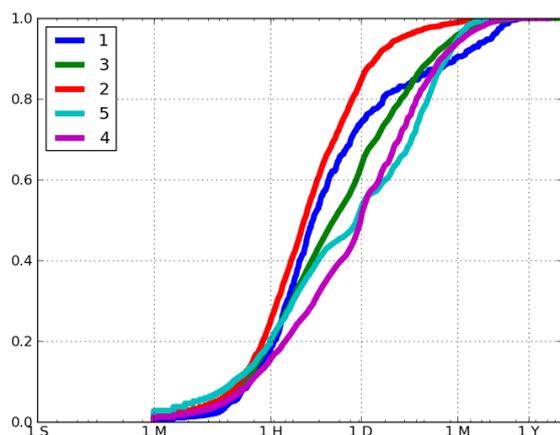
codes is much longer than for the highest two priorities (not shown).

**Incident duration.** Not all classes of incidents take the same time to resolve. Tables 8, 9, and 10 show median incident durations (c.f. Tables 5, 6, and 7). Not surprisingly, Severity 1 and 2 problems are addressed the quickest, with low-severity incidents taking an order of magnitude longer to resolve in many cases. Interestingly, severity-2 problems are generally addressed slightly faster, perhaps because they are frequently less challenging—one imagines the networks are designed to avoid severity 1 problems occurring at all, so their existence may indicate either a chain of failures or a catastrophic event that is hard to recover from. This ordering holds roughly true across the entire distribution of incidents, as shown in Figure 2.

Also unsurprisingly, incidents eventually resulting in no problem found take the longest to resolve. Software inci-

**Table 9: Median duration of incidents (by resolution and class) in customer networks managed by HP.**

Resolution	WAN	LAN	HW	SW	Config	Connectivity	All
Fixed	04:44	24:00	04:56	24:00	19:38	06:34	13:00
No prob. found	08:21	24:00	08:20	7 days	23:45	03:59	21:28
Power restored	03:54	13:18	02:56	17:25	NA	01:39	03:19
Replaced	5 days	N/A	24:00	18 days	17 days	24:00	4 days
3 <sup>rd</sup> Party	10:39	3 days	08:51	N/A	1 day	08:32	11:46
Reboot	18:51	23:10	03:40	N/A	N/A	01:29	03:08



**Figure 2: Distribution of incident duration by severity.**

**Table 11: Classes of incidents in (by severity) HP’s leveraged equipment network.**

Class	Severity			
	1	2	3+	All
WAN	2.9%	12.6%	7%	7%
LAN	2%	23.4%	30%	27%
Hardware	94%	51.4%	47.7%	52%
Software	N/A	N/A	0.2%	0.2%
Configuration	N/A	3.6%	7.1%	6.3%

dents take by far the longest to resolve in general, although configuration incidents are also quite lengthy. WAN incidents, in general, are resolved quickly, unless they require replacing a device, in which case the median duration is a business week (likely due to the human timescales required to dispatch trucks and ship equipment).

### 5.1.2 Leveraged equipment incidents

Leveraged equipment—equipment owned and operated by HP for customer access—accounts for 5% of all network incidents. More than 46% of these devices are firewalls, with the next two largest classes consisting of routers and switches. Table 11 shows the breakdown of these tickets by

**Table 13: Resolution of incidents (by severity) in HP’s leveraged equipment network.**

Resolution	Severity			
	1	2	3+	All
Fixed	4.4%	32.4%	33.5%	30.1%
No prob. found	0%	6.3%	16%	14.1%
Power restored	0.5%	4.5%	1%	1.2%
Replaced	N/A	1%	1%	0.9%
3 <sup>rd</sup> Party	93%	29%	2.4%	11%
Reboot	9%	1%	1%	0.9%

problem class (c.f. Table 5). The severity 1 incidents have a dramatically different breakdown—almost all of them are due to hardware failures, which make sense given we are considering incidents specifically regarding the leveraged hardware.

**Resolution.** Table 12 shows how the four classes of incidents in HP’s leveraged equipment network are resolved. Generally speaking, the distribution is similar to the managed networks, although power plays a less significant role here. The story is different when considering by severity, however. Table 13 shows the relationship between incident resolution and severity. Almost all severity 1, and the majority of incidents in general, are resolved through means not listed here—likely indicating that they resolution was complicated. This fact is not surprising when one considers that leveraged hardware is surely redundant and well provisioned, so any incident is likely to require something more than a simple fix.

**Incident duration.** Hearteningly, Table 14 indicates that the median incident of most classes and severities is resolved more quickly in the leveraged environment than in customer networks. The dramatic exception to the rule is the hardware category, which takes substantially longer. We observe, however, that the median severity-1 hardware failure is dominated by one particular problematic device that failed repeatedly during the course of our study.

Table 15 shows that replacement was much quicker in general for the leveraged equipment, although the few WAN incidents that required replacement to resolve were longer. Similarly, Table 16 shows that “no problem found” incidents

**Table 12: Resolution of incidents (by class) in HP’s leveraged equipment network.**

Resolution	WAN	LAN	HW	SW	Config	All
Fixed	33.5%	33%	25%	N/A	49%	31%
No prob. found	8.2%	11%	18.7%	25%	3.5%	14.1%
Power restored	3.8%	1.3%	0.8%	N/A	N/A	1.2%
Replaced	2.5%	0.8%	0.9%	N/A	N/A	0.9%
3 <sup>rd</sup> Party	4.4%	1%	21%	N/A	N/A	12%
Reboot	1.3%	1.3%	0.8%	N/A	1.4%	0.9%

**Table 15: Median duration of incidents (by resolution and class) in HP’s leveraged equipment network.**

Resolution	WAN	LAN	HW	SW	Config	All
Fixed	11:14	22:47	13:40	N/A	15:17	18:07
No prob. found	02:05	16:09	03:00	24:00	4 days	05:11
Power restored	04:30	03:47	04:32	N/A	N/A	4:32
Replaced	15 days	4 days	24:00	N/A	N/A	48:00
3 <sup>rd</sup> Party	01:31	13:01	5 days	N/A	N/A	4 days
Reboot	08:32	09:00	03:01	N/A	07:32	03:21

**Table 14: Median duration of incidents (by class and severity) in HP’s leveraged equipment network.**

Class	Severity			
	1	2	3+	All
WAN	02:05	4:00	04:43	03:54
LAN	00:34	1:58	23:25	21:52
Hardware	3 days	24:00	16:31	23:17
Software	N/A	N/A	48:00	48:00
Config	N/A	01:05	14:10	12:50

**Table 16: Median duration of incidents (by resolution and severity) in HP’s leveraged equipment network.**

Resolution	Severity			
	1	2	3+	All
Fixed	03:46	04:00	19:09	18:07
No prob. found	01:21	01:21	05:29	05:11
Power restored	04:32	04:32	04:40	04:32
Replaced	N/A	01:31	4 days	48:00
3 <sup>rd</sup> Party	3 days	5 days	2 days	4 days
Reboot	N/A	00:40	03:21	03:21

Ticket severity	% tickets with matching call record
1	42%
2	25%
3	49%
4	55%
5	36%

**Table 17: Ticket severity versus customer calls.**

were resolved much more quickly as well.

## 5.2 Customer-reported incidents

In addition to incident tickets, our database also contains call tickets, registered whenever a customer calls for support. Call records and incident tickets are subsequently linked, and thus we can distinguish between those incidents that also incur customer calls and those that do not. Overall, this split is roughly even.

One might guess that customers will be most vocal about the most significant incidents, since these events are most likely to impede critical business services. However, this is not the case.

Table 17 documents the fraction of tickets at each severity level that have associated call records. We see that in fact severity 1, 3 and 4 incidents incur roughly the same incidence of customer reporting. Indeed, it is only severity 2 incidents that see a smaller number of customer calls. We hypothesize that these tickets correspond to partial connectivity failures, during which redundant links are available, and thus customers may not explicitly notice the failure themselves. This is supported by the observation that the problem type *Connectivity* is used almost exclusively for customer-

**Table 18: Ticket severity versus user impact.**

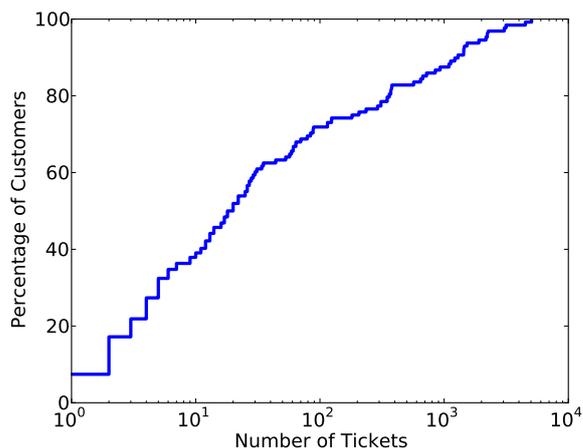
Ticket Severity	% tickets with 1 user affected	% tickets with 100+ users affected
1	31%	48%
2	58%	4.7%
3	77%	4.5%
4	68%	4%
5	92%	0.6%

initiated connectivity problems, and almost never appears in severity 2 tickets.

Another hypothesis is that perhaps users are likely to call with spurious complaints. We therefore looked at the fraction of incidents whose ultimate resolution is coded as *No Problem Found*. We found that such tickets represent only 6% of all tickets, and a similar fraction of unresolved problem minutes, if there is an associated call record. But when there is no associated call record, tickets with *No Problem Found* represent 14% of all tickets, and 18% of unresolved problem minutes. Thus, user complaints are in fact a more reliable source of “ground truth” than automated systems. This is perhaps not surprising in retrospect, given the noisy nature of low-level network event systems, and the fact that “customers” in this setting are not typical end-users, but experienced IT managers. (The possibility does exist that the increase in unresolved problem minutes for no-call-record tickets represents a lower priority for solving apparently-spurious problems when the customer is not complaining.)

The other significant difference between customer-driven and manager-driven tickets is the diversity in problem types. While the problem type schema is able to code for a larger number of distinct causes, we find that 23% of customer-driven tickets are coded with a problem type of *Other* (constituting over half of unresolved problem minutes for such tickets), while for manager-driven tickets the fraction is only half as large (and unresolved problem minutes is smaller still). Thus, customers are simply bringing a far broader class of problems, and problems with a larger distribution of times required to fix them. (Indeed, these “other” problems have a median unresolved duration of more than 2 days, compared with 11 hours for equivalently coded tickets in the manager-driven category. The tail can be larger still, depending on subcategory.)

Our data set also includes some information about the number of users affected by a failure. We suspect that, in many cases, this data is unreliable (for example, it might reflect a rough estimate or just a default choice), although in some cases it seems consistent with the text in incident tickets. Table 18 shows that there is a clear relationship between severity and estimated user impact, in that the severity 1 and severity 2 tickets are more likely to report a large number of affected users.

**Figure 3: CDF of tickets per customer.**

Problem type	25 <sup>th</sup>	50 <sup>th</sup>	95 <sup>th</sup>	mean	std. dev.
	%ile	%ile	%ile		
break/fix	0	0	33	7.5	14
connectivity	0	0	16	2.8	10
failure/down	0	0	44	7.5	17.6
error message	0	0	12	2.4	8.9
other	0	0	71	13.4	23.4

**Table 19: Variation of problem type frequency (percentage of tickets) among customers.**

### 5.3 Variations by Customer

Historically, network measurement studies have covered only one network, whether it be an ISP, campus, or data center, or at most a small set of networks. Because of our unique opportunity to compare data for thousands of customer networks, we can explore whether single-network studies of enterprise networks could provide valid results.

We begin by looking at how frequently customers have network related incidents. Figure 3 is the CDF of tickets per customer. While the median customer has approximately 20 tickets, 10% of customers have over one thousand. The large variation in tickets per customer is likely due to several reasons centering around the size of a customer’s network and the level of service HP ES provides.

We next examine the variation in problems types by customer (note that a given “customer” may in fact comprise multiple managed networks). For each customer we calculate the fraction of each problem type found in incident tickets and then compute the 25<sup>th</sup>, 50<sup>th</sup>, and 95<sup>th</sup> percentiles for this distribution (in addition to the mean and standard deviation) as shown Table 19.

In fact, we find that this distribution is far from uniform. The 25<sup>th</sup> percentile for the five most common problem types is consistently zero; for three of the problem types even the median is zero. Combined with the large standard devia-

Severity	25 <sup>th</sup> %ile	50 <sup>th</sup> %ile	95 <sup>th</sup> %ile	mean	std. dev.
1	0	0	4.3	0.85	1.5
2	0	3.1	27.5	6.4	9.9
3	63.5	84	97.5	74	23.2
4	1.2	5.7	41	10.3	15.4
5	0.5	3.3	59	8.6	14.9

**Table 20: Variation of severity frequency (percentage of tickets) among customers.**

tions for each type, this implies that these top problem types are not evenly distributed among customers. Thus, for some customers connectivity errors may be 50% of all problems, while for others than may not have any such problems.

One potential explanation for the large variation in problem types by customer could be that each customer experiences its own unique set of problems. To determine if this is the reason we looked at how few customers were required to achieve total coverage of all problem types. Figure 4(a) shows an upper bound on this result; a greedy heuristic was used when picking customers to make the problem tractable. Immediately we see that less than five percent of all customers are required to achieve coverage of 70% of all problem types, and complete coverage is possible with less than twenty percent of all customers. This result clearly shows us that complete problem type coverage is possible with only a small set of all customers. However, it does not tell us if we can use any set of companies or only one specific set.

To determine if we can use any set of customers to achieve the same results, instead of an optimal set, we once again plot percentage of customers vs. percentage of problems covered. But we randomly pick ten thousands sets of the correct size and plot the median percentage of problems covered, Figure 4(b). Here we see that we can achieve 50% coverage of problem types by with only 20% of all customers, but to achieve complete coverage of all problem types requires almost every single customer. Thus, there are in fact some companies have unique problems.

We also explore the question of variation within customers with respect to problem severity. Are high-severity events evenly distributed across customer sites, or do certain customer sites require more attention? Table 20 shows the percentage of incidents per customer, broken down by severity. As expected, severity three issues continue to be the most common across all customers. Moreover, while most other problems are uncommon (the median fraction of severity two, four, and five incidents is less than 6% and half of customers experience no severity 1 problems), there is significant variation for severity two, four and five with a standard deviation double the mean. Thus, while severity one problems are uniformly rare (even at the 95<sup>th</sup> percentile, only a small fraction of any customer’s problems are severity one), the other severity levels can vary significantly from one customer to another.

Given the variations we observed, we conclude that a study that focuses on only a few enterprise customer sites will have a high chance of mis-characterizing the breadth of customer problem types and severities.

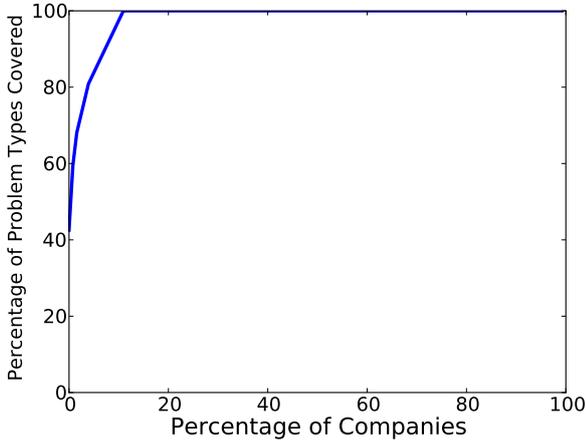
## 5.4 Change

In a well-functioning network, one might expect the majority of incidents to revolve around restoring hardware or services to their previously functioning state. However, some incidents will require resolutions that in fact do not result in restoring to a previously functioning state, because the previous state was in fact not functioning appropriately. For example a firewall rule was discovered to block an important port. Five percent of all incidents required a network change be made to resolve the incident. Figure 5 lists incidents by category with the percentage of all incidents in each category. The figure contains two tables, one for incidents requiring a network change, and one for the remaining incidents. We indicate the median duration of the incidents requiring change (the median duration of the others, since it is more than 96% of the incidents, is approximately the same as the overall data presented earlier). Due to their importance, we separately highlight severity-1 incidents.

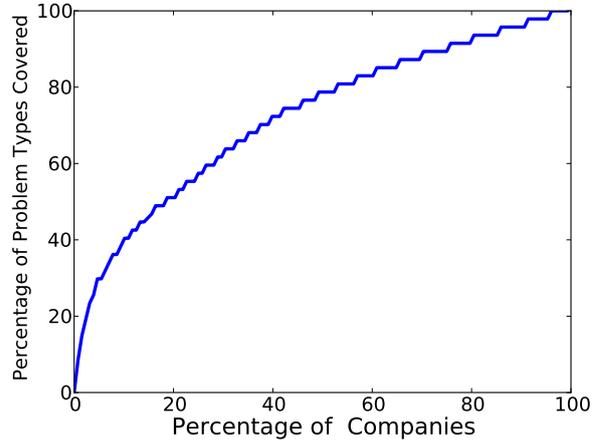
As shown in Figure 5(b) the most common category is network software, which is also over-represented when compared to incidents not requiring changes. On the other hand only a tenth of a percent of all incidents are classified as configuration issues. This is puzzling as one might expect most issues requiring changes to be the fault of incorrect configuration. We also notice that a significant number of events fall into the network hardware sub-category. This might lead one to believe that some incidents may require new hardware supporting newly available features, or that specific topologies are unable to provide the redundancy required.

Looking at the median durations from Figure 5(b), the large median duration of the network software sub-category jumps out. We dug deeper to find out why this was so long. Of the 44% of all incidents requiring a change that are of sub-category “network software,” one customer is responsible for 88%. This one company’s incidents dominates the network software sub-category, causing it to have such a large median duration. For the remaining sub-categories, the median durations for network hardware and LAN are a few hours longer than normal but the median durations for “connectivity,” “WAN,” and “configuration” are actually shorter.

In general, incidents requiring changes have median durations that are hours longer than those incidents not requiring a change. This seems reasonable, given that operators must decide what the correct behavior should be, determine how best to implement the change, and finally execute the change. If we look at resolution codes (not shown), we find that reverting back to an original state is done for 1.5% of incidents; perhaps these incidents require larger changes than operators are comfortable making on the fly. Further, only 1.8% are resolved by changing a configuration. Instead, the



(a) Upper bound



(b) Expected value with randomized selection

**Figure 4: Percentage of customers needed to cover various fractions of problem types.**

majority of resolutions belong to fix/repair/correct.

## 6. DISCUSSION

Analyzing our data set, while fruitful, was also filled with difficulty from various sources. In this section we discuss several limiting factors in building a holistic view of network incidents, and potential ways to improve such issues.

### 6.1 Missing Information

Our data analysis provided a relatively robust characterization of enterprise incidents including causes, resolution, perceived severity, and category. However, there were pieces of information that we could not extract from our data sets. Ideally we would have liked to be able to provide additional information such as technical severity of each incident or at least some level of verification that an impacting incident occurred as well as the reliability of devices by functionality and model. In order for us to have been able to explore these areas we would have needed additional data sources that, while unavailable to us, are collectible in theory.

Early in our work we discovered our limited visibility into each network's topology. We quickly found that our limited visibility left us unable to come up with a set of rules or heuristics to accurately capture the purpose or importance of most interfaces in our data set. This lack of information made it very difficult to verify the impact of incidents, as well as to speak to reliability in terms of functionality.

We struggled to develop a methodology to overcome our limited visibility into network topology. However, without reliable information as to a link's functionality, or if there were additional back-up links in the unmanaged topology, we found ourselves at an impasse. Thus we were left with only one metric for evaluating impact, the customer ticket's severity value. Given topology information for the unmanaged parts of each network, we would have been able to determine link functionality as well as redundancy information. With these two pieces of information we could have then developed a better methodology to understand the im-

Category	% of tickets	Sev 1
Network Hardware	38%	0.7
Network Software	13%	0.03
LAN	12%	0.2
Connectivity	11%	0.14
Hardware	6.8%	0.03
Configuration	6%	0.03
WAN	4.7%	0.2

(a) Without change

Category	% of tickets	med. dur.	% Sev 1
Network Hardware	1.2%	8:53	0.02
Network Software	1.7%	6 days	0.01
LAN	0.2%	35:22	0.01
Connectivity	0.2%	2:16	0.03
Hardware	0.1%	19:02	0.002
Configuration	0.1%	14:42	0.007
WAN	0.1%	15:01	0.002

(b) Accompanied by change

**Figure 5: Incidents by category, segregated according to whether or not they required a network change to resolve the incident.**

pact of incidents.

A different approach to understanding network impact could have come through direct measurements. The most straightforward measurement would be to use SNMP to record the value of various counters like incoming and outgoing packets on each interface. Such information would have given us the ability to confirm link failures. A more familiar method would have been to ping managed devices from more than just a central facility. These additional pings, originating from inside of a client network, could potentially allow us to identify exactly where connectivity was broken, which can be especially difficult in large L2 topologies with redundancy. A more detailed method for measuring impact would be to deploy netflow, a protocol for randomly sampling incoming traffic on each port. Netflow would allow us to not just determine the rate of traffic, but also be able to detect failures that prevented specific types of traffic from being transmitted. However, even with netflow enabled, we would be unable to verify some events, such as “lost password.”

## 6.2 Difficulty in merging data sources

In Section 4.3 we attempted to match our incident tickets with raw syslog. In the end we were only able to match 1% of all tickets to a syslog event. The most significant hurdles to successfully matching syslog to incident tickets is that only 25% of all tickets listed a recognized device name. While not all tickets were expected to list a recognizable device name, we suspect that many more should have. It would be easy to blame the lack of recognizable device names on network technicians, but we believe the root of the problem stems from a discrepancy between how syslog messages are processed and how they are designed.

Like other large service providers the volume of syslog messages that HP receives each day requires the use of automated filtering systems. However, if one looks at the structure of syslog messages, the majority of the relevant information is contained in the human-readable section of the message. Because these messages are intended to be human-readable, the format and location of important information can change, making it difficult to build systems that can analyze syslog in real-time [9]. When technicians do find a relevant syslog message, the excessive message length encourages them to copy only the most relevant contiguous parts into the ticket, which often excludes the device name. Hence we believe syslog messages should no longer be structured for human readability, but instead should be structured in a way that makes them optimized for machine parsability.

Along with restructuring the format of messages, we also believe it is important to reconsider how various elements in each message are named, the two most important being device and interface names. Most syslog daemons attempt to resolve the IP address that sent the syslog message to a hostnames, instead of having the network device send a FQDN. This is problematic if a network device has multiple IP ad-

resses that resolve to different names or if IP addresses are moved without names be changed. Instead devices should send a FQDN or other unique identifier in each message and syslog daemons should record IP addresses, in addition to the resolved names. We also believe that instead of sending interfaces names like *GigabitEthernet0/0/0* or *GigE0/1* syslog should send MAC addresses, to prevent confusion if line cards are moved, and to avoid the need to parse different formats.

Beyond changing formats of messages and how elements are named, we also believe that an additional field would be helpful, an event ID. In many cases a single event will impact several protocols at higher levels. Currently, it is difficult to determine if multiple separate incidents are occurring at the same time, and which, if any, of the messages represents the root cause of each incident. However, by using shared event IDs it would be straightforward to observe both the full impact of a single incident as well as identify the potential root event. An even more sophisticated system would allow event IDs to be shared across multiple devices; e.g., when a cable is severed both sides would send link-down messages with the same event ID. These distributed IDs would make it easier to find all messages relating to one event, especially during the initial debug phase, and operators could simply copy event IDs directly into incident reports making it very easy to back reference.

## 7. CONCLUSION

In this paper we have described a preliminary characterization of failures as seen in managed enterprise networks. Among our findings, we show how low-level network event data is unable to capture the full range of problems handled by ESPs, that high severity errors are dominated by connectivity problems with third-party ISPs and that trouble tickets are dominated by lower-severity problems. Finally, we document that there is significant variation in the prevalence of different problem types and severities across different customers.

## Acknowledgments

The authors would like to thank Greg Hubbard, Brian Pearse, and Craig Kiser for providing us with access to HP ES data as well as helping us to understand its uses and meaning.

## 8. REFERENCES

- [1] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. In *SIGCOMM*, 2007.
- [2] B.-Y. Choi, S. Song, G. Koffler, and D. Medhi. Outage analysis of a university campus network. In *ICCCN*, 2007.
- [3] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot. NetDiagnoser: Troubleshooting network

- unreachabilities using end-to-end probes and routing data. In *Proceedings of CoNEXT*, 2007.
- [4] N. Duffield. Network tomography of binary network performance characteristics. *IEEE Transactions on Information Theory*, 52(12):5373–5388, 2006.
- [5] P. Gill, A. Greenberg, N. Jain, and N. Nagappan. Understanding network failures in data centers: Measurement, analysis, and implications. In *SIGCOMM*, 2011.
- [6] Y. Huang, N. Feamster, and R. Teixeira. Practical issues with using network tomography for fault diagnosis. *Computer Communication Review*, 38(5):53–57, October 2008.
- [7] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren. IP fault localization via risk modeling. In *Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 57–70, Boston, MA, May 2005.
- [8] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, Y. Ganjali, and C. Diot. Characterization of failures in an operational IP backbone network. *Transactions on Networking*, 16(4), 2008.
- [9] T. Qiu, Z. Ge, D. Pei, J. Wang, and J. Xu. What happened in my network: mining network events from router syslogs. In *IMC*, 2010.
- [10] T. Qiu, Z. Ge, D. Pei, J. Wang, and J. Xu. What happened in my network: mining network events from router syslogs. In *IMC*, 2010.
- [11] A. Shaikh, C. Isett, A. Greenberg, M. Roughan, and J. Gottlieb. A case study of ospf behavior in a large enterprise network. In *IMW*, 2002.
- [12] G. Tan, M. Poletto, J. Gutttag, and F. Kaashoek. Role classification of hosts within enterprise networks based on connection patterns. In *USENIX ATC*, 2003.
- [13] A. A. Theophilus Benson and D. A. Maltz. Network traffic characteristics of data centers in the wild. In *IMC*, 2010.
- [14] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage. California fault lines: Understanding the causes and impact of network failures. In *Proceedings of the ACM SIGCOMM Conference*, New Delhi, India, Aug. 2010.
- [15] T. Wang, M. Srivatsa, D. Agrawal, and L. Liu. Spatio-temporal patterns in network events. In *Co-NEXT*, 2010.